System Configuration

Paul Mooney
Chief Software Architect, Microsoft MVP

@daishisystems | www.insidethecpu.com

Overview

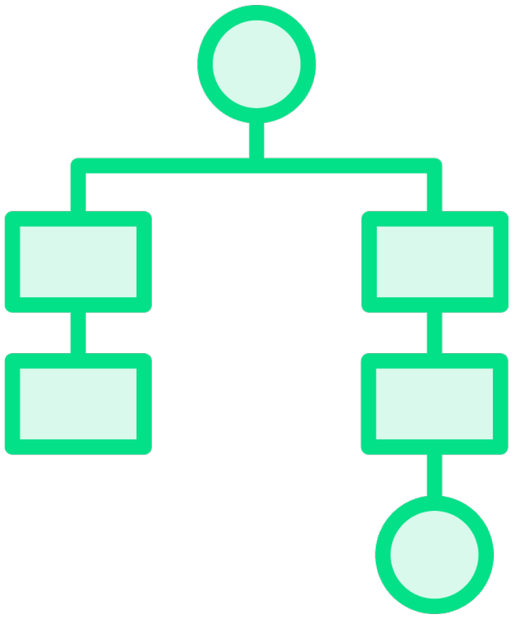
Coming Up

- Mitigate directory listing vulnerabilities
- Disable directory listings, restrict access, create index files
- Remove unnecessary files, limit exposure
- Remove sensitive HTTP response headers
- Implement security measures: least privilege, secure error handling
- Isolate development, change control, asset management

Demo

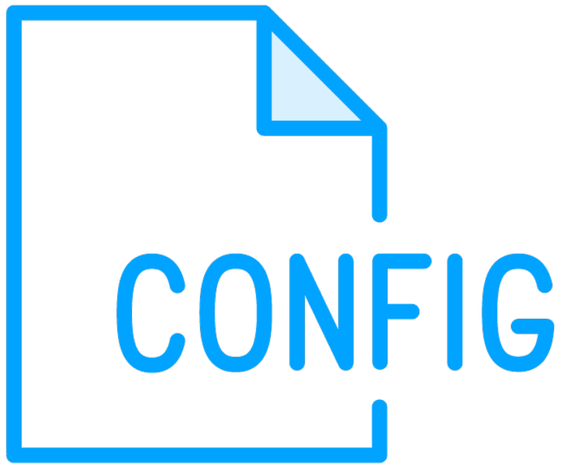
Effective Disabling Directory Listing

- Mitigate potential vulnerabilities diligently
- Unintentional directory listings: risk, sensitive files
- Disable directory listings to prevent exposure



Restricting Directory Listing

- Disable directory listings in web app
- Implement access restrictions, prevent unauthorized access
- Create index files, hide directory listings



Remove/Disable What You Don’t Need

- Eliminate unnecessary functionalities and files
- Remove test code and unused functions
- Limit developer-layer access to these elements
- Pay attention to HTTP response headers
- Remove headers that disclose sensitive information
- Avoid revealing OS, web server, and programming language details

Remove/Disable What You Don’t Need

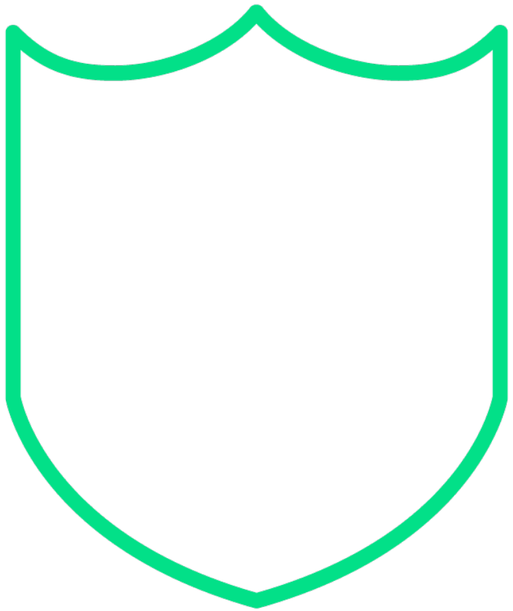
```
Etag: W/"75a11da44c802486bc6f65640aa48a73"
Referrer-Policy: no-referrer-when-downgrade
Server: GitHub.com
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
Vary: X-PJAX, X-PJAX-Container, Turbo-Visit, Turbo-Frame, Accept-Encoding, Accept, X-
Requested-With
X-Request-With: "Go Vulnerable Framework 1.3"
```

Remove/Disable What You Don’t Need

```
w.Header().Set("X-Request-With", "Go Vulnerable Framework 1.3")
```

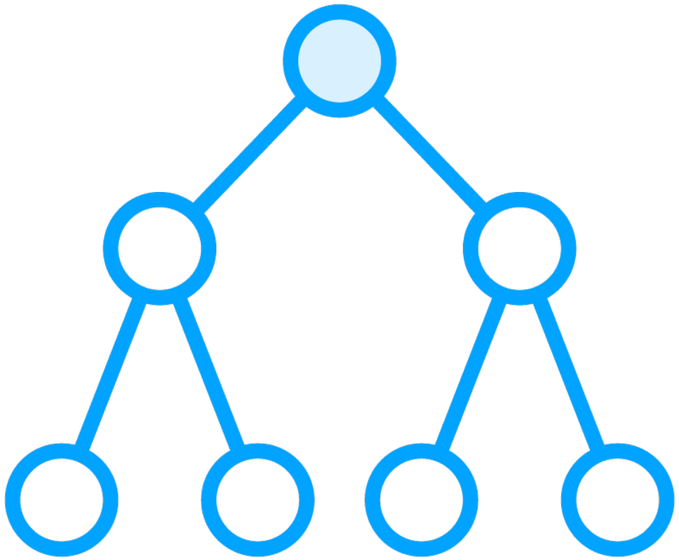
Remove/Disable What You Don’t Need

```
w.Header().Set("Access-Control-Allow-Methods", "POST, GET")
```



Implement Better Security

- Prioritize security and least privilege
- Securely handle exceptions, prevent disclosure
- Refer to Error Handling and Logging modules for more details



Prevent Directory Structure Disclosure

- Create a robots.txt file
- Include a "Disallow" directive for all directories and files by default
- Explicitly allow access to specific directories and files that should be accessible

Configuring robots.txt

```
User-agent: *
Allow: /sitemap.xml
Allow: /index
Allow: /contact
Allow: /aboutus
Disallow: /
```

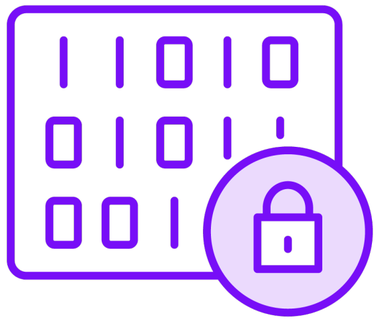
Best Practices

- Isolate development environment from production network
- Implement access controls and security measures
- Use software change control system effectively

Asset Management System

- Overview of Asset Management Systems
- Optimize asset performance, evaluate security
- Include components and software as assets

Asset Management System Implementation



Information Security
AMS Scope
Security Policy

Structure security
Classify assets
Assess risks

Risk management
Identified Risks
Applicability statement

Train Staff
AMS
Performance
Maintain AMS

Summary

System Configuration

- Secure coding
- Stay up-to-date
- Directory listings vulnerability
- Remove unnecessary code and files in production
- Restrict sensitive HTTP response headers
- Restrict supported HTTP methods
- Implement Cross-Origin Resource Sharing (CORS)
- Adhere to the principle of least privilege
- Protect directory structure